Removable Security Device

Product Evaluation Guidelines

Smart Card Reader, USB Security Token, PC USB SIM Card Reader

November 2002

Revision 1.1

Contents

Tables

1. Device Validation Recommendations

This document outlines evaluation procedures for removable hardware security devices (smart card readers, USB security tokens, PC SIM card type readers) for use with notebook computers. The intent of these guidelines is to provide guidelines for functionality and interoperability working cooperatively with vendors of removable hardware security devices to ensure quality products are delivered to the mobile PC industry. Vendors are encouraged to identify and correct any anomalies and then submit updated software and/or hardware to Intel for re-verification. Intel will publish evaluation results periodically to include those products who meet the expectations defined herein.

The focus of the tests outlined is to check the interaction of this hardware security device with the notebook PC. These guidelines do not define tests for any types of crypto functions of the products. Furthermore, these tests are not intended to supercede or replace any other industry testing efforts, nor replace any customer evaluation process.

Only devices connecting to USB or PCMCIA are to be tested. Products connecting to serial or other legacy ports will not be tested.

1.1. Functional Testing

Functional testing can be conducted by vendors with their own products using standard notebook PCs from a variety of vendors. Intel may reproduce testing results in our own lab environment prior to publishing any results.

Tests are expected to be performed under both Windows 2000* and Windows XP* unless otherwise noted. Each Banias reference system will be refreshed to a pristine environment before vendors test products. This is done so that no interaction problems can be encountered by software which may be left by a preceding test of another hardware product.

All smart card reader testing should be done with Gemplus, Infineon or Schlumberger smart cards as support for these cards is included with the operating system. In some cases, the test area is not applicable to a particular product. For example, smart card readers are not expected to perform CAPI testing as the tests apply only to the smart card itself and not the reader. Those few exceptions are noted in the outline below where appropriate.

All removable security devices should have all previous certificates removed / erased prior to starting a new test station. Vendor tools provide a method of deleting certificates from the device and/or formatting the device to a fresh state.

1.2. Platform Configuration

Intel testing will be conducted on a reference development system based on the Intel Banias processor. The platform is defined in the following table.

Platform	Intel Banias Customer Reference Board
Memory	128MB
Hard disk	Size: 30GB (primary) A second physical disk contains GHOST images of Windows* XP and Windows 2000*.
Connectors (USB, PCMCIA, …)	USB, PCMCIA
Software Applications Installed	McAfee* Virus Scan*

Vendor testing can be done with a variety of standard notebook computers based on the Mobile Intel® Pentium® 4 processor. Systems that Intel has used for such verifications include, but are not limited to:

- IBM Thinkpad® T23 (Intel Mobile Pentium III Processor-M)
- IBM Thinkpad® T30
- IBM Thinkpad® R31
- Compaq Evo N600C
- Compaq Evo N800C
- Toshiba Tecra 9000
- Toshiba Tecra 9100
- Dell Latitude™ C610
- Dell Latitude™ C840

2. Overview Checklist Testing

This section provides simple overview checklist testing for each product. Before evaluating devices in this section, ensure that the platform operating system configuration has been restored to its original installation state to eliminate other software or hardware interactions which may skew evaluation results.

2.1. Device Hardware Configuration

Removable security devices should connect to the notebook computer either using the PCMCIA slot or by the USB connector. Because serial ports are considered legacy connections that should soon start to disappear from PCs, such connections should not be used for removable security devices.

PCMCIA devices can use either the PCCard-16 or PCCard-32 (Cardbus) interface type. Use of a PCMCIA removable security device must not preclude use of other PCMCIA devices that may be blocked or hindered by the security device protruding too far.

2.2. Software Install, Configuration, and UnInstall

Configuration of these devices should be an automatic process initiated when the device is plugged into the system causing the Windows device installer to search for device driver software for the new removable security device.

If the device is plugged into either USB or PCMCIA, then Windows should search for the proper device driver. If it is not found as part of the default Windows installation, Windows checks media already in the CDROM drive. If not found, the user will be prompted to insert the right media with the device driver for the removable security device. Inserting the CDROM should allow the Windows device installer to find the correct INF file in the root of that media to enable installation of software for the hardware device. The INF file should provide all the information necessary to install all the required support pieces for the removable security device including the device driver, MS-CAPI CSP, PKCS#11, and other vendor specific utilities and configuration. The vendor may alternatively provide software installation through a "setup" utility that installs all necessary software for the device. Although it is preferred to have an INF file in the root directory of the installation media, the user setup software installation is also desirable.

A software installation utility should be provided on the security product's CDROM media provided by the vendor. Inserting the CDROM should automatically start (autorun) the installation program to walk the user through the installation steps. If software installation is done prior to inserting / plugging in the removable security device, then the Windows device installer should properly install and configure the device driver so that the Windows device installer should not request further interaction when the device is plugged in.

The software installation utility should provide a simple and easy to use interface. Complicated configuration options should not be required. Software should install and work properly without requiring the user to reboot the system. Device drivers should be dynamically removed by the OS when the device is unplugged such that the device driver does not show up in the Windows Device Manager unless the device is attached. This means that the device driver and support software should be capable of dynamic installation rather than static installation.

The software installation process should configure a means to uninstall this software through the Control Panel "Add/Remove Software" menu. The uninstall process should cleanly remove all software for the device. Removable security devices that have device driver support bundled with the operating system are not expected to have provisions to uninstall that software support.

A software utility should be included with the product to allow the user to examine and manipulate the contents of the device. This utility may provide a means to change the device password, remove certificates, and/or refresh/format the device.

2.2.1. Installation With Multiple Operating Systems

The removable security devices should be tested under both Windows 2000 and Windows XP operating systems. As each operating system has unique characteristics, it is important to test under both operating environments.

Windows 2000

1. With system running Windows* 2000, install and configure the removable security device.

2. Upon device attach, the “New Hardware Found” dialog box appears.

3. During installation, no part of the software should cause any sort of a warning message from the operating system concerning lack of “certification” or “driver signing” giving the impression that the software has not been tested and approved by Microsoft.

4. Windows is able to complete software installation without further user input

5. Installation should not require system reboot.

6. Using vendor application, read configuration data from removable security device.

Windows XP

1. With system running Windows* XP*, install and configure the removable security device.

2. Upon device attach, the “New Hardware Found” dialog box appears.

3. During installation, no part of the software should cause any sort of a warning message from the operating system concerning lack of “certification” or “driver signing” giving the impression that the software has not been tested and approved by Microsoft.

4. Windows is able to complete software installation without further user input

5. Installation should not require system reboot.

6. Using vendor application, read configuration data from removable security device.

2.3. Cryptographic Software Interface Support

Support for Microsoft CAPI should be either a standard part of the operating system or installed with device drivers for the hardware. The same is true for the PKCS#11 support: it should ship as a standard part of the operating system, or should be installed with hardware device drivers.

Support for CAPI and PKCS#11 apply only to devices that deliver cryptographic functions by themselves. This support is not to be tested for smart card readers, but does apply to USB security tokens or other such devices.

CAPI CSP Support

Software support should be delivered with the security device that is in compliance with the Microsoft CAPI v2.01 interface with a CSP certified and signed by Microsoft. Verification can be done using the CSPTESTSUITE software utility from Microsoft. Additional information regarding this test is available through requests to: `cryptoapi@disuss.microsoft.com`

1. If you have not already done so, install the removable security device and it’s software support.

2. Click **Start**, click **Run**, type **cmd.exe**, and then press ENTER.

3. From the command prompt, type **CSPTESTSUITE**. This utility will report the name of the registered CSP files.

4. Find the name of the CSP for your device, and run CSPTESTSUITE –c "*cspname"*-f 10 and press ENTER where cspname is the name of your CSP. Quotes should be included in the command line parameter. The –f 10 parameter will cause CSPTESTSUITE to test the "CryptGenKey" API.

PKCS#11 Support

Because there is no "registration" mechanism defined for PKCS#11, the DLL name providing support for the hardware security device must be known. The PKCS#11 support should be written to the Cryptographic Token Interface Standard v2.01 or later as defined by RSA*.

1. Install the removable security device and it's software support.

2. Click **Start**, click **Run**, type cmd.exe, and then press ENTER.

3. Use the **PKCSVER** utility to retrieve the version number for the PKCS#11 DLL by typing PKCSVER followed by the name of the DLL with it's pathname, and then press ENTER.

4. The PKCS#11 version number should be v2.01 or later.

2.4. WHQL Tests

Microsoft WHQL tests are available for download from the Microsoft Developer web site. It is expected that all removable security device vendors have passed these WHQL tests. No WHQL specific tests should need to be duplicated. However, it is important to verify whether the specific product meets WHQL compliance by checking the Microsoft supported products lists.

Background: What do Logo'd Products Offer for End Users?

The "Designed for Windows" logo can play a key role in reducing total cost of ownership for organizations. The Windows logo criteria is one of Microsoft's key vehicles in communicating to software developers how to incorporate Zero Administration for Windows (ZAW) initiative features into their applications in preparation for the next releases of Windows.

Businesses that use products meeting the Designed for Windows logo criteria stand to gain the following benefits:

- **Lower support costs:** Logo'd applications follow standard Windows look-and-feel guidelines. That helps users get up to speed without phoning your helpdesk.

- **Help manage "DLL Hell":** Logo'd applications do not overwrite applications or uninstall key components.

- **Support mixed Windows environments:** Logo'd products work in Windows environments to help manage the mix of Windows desktops in an organization

- **Proper use of operating system:** Logo'd products make proper use of the Windows registry and other key operating system files.

- Compliance with the Americans with Disabilities Act and other equal rights legislation: Logo'd applications meet usability standards for a wide range of people.

A check should be made to determine if this product is listed in the Microsoft WHQL compliance list. This Microsoft list is found at:

http://www.microsoft.com/windows/catalog/wcbody.asp?subid=22&catid=6bc37356-d760-4736-9a32-baf3a2c3f34e

2.5. Overview Guidelines Checklist

The following checklist summarizes the important aspects of a removable security device to meet the guidelines defined in this document. All devices should meet these recommendations in each case.

This short list of evaluation points is not intended in any way to be a comprehensive validation of any particular product. It is however intended to represent general "mobile friendliness" of devices as described in this set of vendor recommendations.

Line items that indicate a "NA" for the compliance metric should only be marked if the device does not fall into that category. For example, a PCMCIA device should not attempt to measure compliance against USB recommendations. Smart card readers that do not provide built-in crypto functions are not required to support MS-CAPI or PKCS#11.

Vendor Representative		
Product Name / Number		
Device Driver Name / Number	Windows 2000:	Windows XP:
Bus Interface (USB, PCMCIA)		

Table 1: Overview Guidelines Checklist

Guideline	Functions as Expected (Windows 2000)	Functions as Expected (Windows XP)
1. Device connection via USB or PCMCIA	Yes No	Yes No
2. All software entities that should be installed to a user's hard drive should be described in an INF description file shipped with these collaterals on the root directory of the product software support CDROM. Alternatively, software installed through a "setup" utility.	Yes No	Yes No
3. PCMCIA devices are automatically detected when inserted for the first time, and either automatically configured, or prompt the user to	Yes No NA	Yes No NA

Guideline	Functions as Expected (Windows 2000)	Functions as Expected (Windows XP)
insert media with software support. Software needed for this device is installed with minimal user interaction.		
4. USB devices are automatically detected when inserted for the first time, and either automatically configured, or prompt the user to insert media with software support. Software needed for this device is installed with minimal user interaction.	Yes No NA	Yes No NA
5. All software collateral should be delivered on CD-ROM. When inserted, the CDROM should "autorun" for installation. And/or provides a simple setup utility for software installation.	Yes No NA	Yes No NA
6. Device drivers should be fully functional after simple installation *without restarting the PC.* (dynamic device driver rather than static.)	Yes No	Yes No
7. A mechanism for uninstalling the software should be provided after installation so the user can remove software through the Control Panel "Add/Remove Software" menu	Yes No NA	Yes No NA
8. User configuration and installation should be simple (ease of use): includes a user-friendly, simple to understand interface.	Yes No	Yes No
9. All software collateral should be supported in both Microsoft Windows 2000* and Windows XP* operating environments.	Yes No	Yes No
10. All software device drivers for removable security devices should have been tested and signed by Microsoft for use with Windows 2000 and Windows XP. Device drivers should be WHQL certified. Installation should occur without warning by the Windows device installer.	Yes No	Yes No
11. Device found on Microsoft WHQL compliance "good" list? (Check the Microsoft web site for the latest list.)	Yes No	Yes No
12. Device found on Microsoft WHQL "Designed for Windows XP" device list (Check the Microsoft web site for the latest list.)	NA	Yes No

Guideline	Functions as Expected (Windows 2000)	Functions as Expected (Windows XP)
13. Software should include a utility to read configuration information from the removable security device	Yes No	Yes No
14. Support should be provided for the removable security device in compliance with the Microsoft CAPI v2.01 interface with a CSP certified and signed by Microsoft.	Yes No NA	Yes No NA
15. Support should be provided for the hardware security device in compliance with the PKCS #11 interface Cryptographic Token Interface Standard v2.01 or later as defined by RSA*.	Yes No NA	Yes No NA

Comments:

3. Application Testing

Application testing should be done with real world applications to ensure the removable security device functions in an expected fashion. Although testing in this section is not exhaustive, it provides a representative sample of real application use of the removable security device.

1. Ensure that the platform has been restored to its original state.

2. Install and configure device and software for the removable security device

3. Run McAfee Virus scan to ensure no virus is installed during that process.

Smart card readers are expected to perform these tests to ensure the reader provides the pathway to the smart card itself. Standard smart cards from Gemplus, Schlumberger, and/or Infineon can be used for testing smart card readers.

3.1. Simple Windows Logon Configuration Testing - Background

Smart cards directly implement a two-factor authentication policy, and indirectly permit data confidentiality, data integrity, and non-repudiation for multiple applications, including domain logon, secure mail, and secure Web access. Smart cards rely on the public key infrastructure (PKI) of Windows 2000 and Windows XP. Smart cards can only be used to log on to domain accounts, not local accounts. When a password is used to log on interactively to a domain account, Windows 2000 Server and Windows XP Professional use the Kerberos V5 protocol for authentication. If a smart card is used for logon, the operating system uses Kerberos V5 authentication with X.509 v3 certificates unless the domain controller is not running Windows 2000 Server.

To initiate a typical logon session, a user must prove the user's identity to the KDC by providing information known only to the user and the KDC. The secret information is a cryptographic *shared key* derived from the user's password. A shared secret key is symmetric, which means that the same key is used for both encryption and decryption.

To support logging on by using a smart card, Windows 2000 Server implement a public key extension to the Kerberos protocol's initial authentication request. In contrast to shared secret key cryptography, public key cryptography is asymmetric; that is, two different keys are needed — one to encrypt, another to decrypt. Together, the keys needed to perform both operations make up a private/public key pair.

When a smart card is used in place of a password, a private/public key pair stored on the user's smart card is substituted for the shared secret key derived from the user's password. The private key is stored only on the smart card. The public key can be made available to anyone with whom the owner wishes to exchange confidential information.

In the public key extension to the Kerberos protocol, the client encrypts its part of the initial Authentication Service Exchange (AS Exchange) with the private key and passes the certificate to the KDC. The KDC encrypts the user's logon session key with the public half of the user's key pair. The client then decrypts the logon session key by using the private half of the key pair.

Initiating a smart card logon session involves the following process:

1. The user inserts a smart card into a card reader attached to the computer.

2. The insertion of the card signals the SAS just as pressing CTRL+ALT+DEL signals the SAS on computers configured for logging on using a password.

3. In response, Winlogon dispatches to MSGINA, which displays a modified logon dialog box. In this case, however, the user types only the personal identification number (PIN).

4. MS-GINA sends the user's logon information to the LSA just as it does with a logon session using a password.

5. The LSA uses the PIN for access to the smart card, which contains the user's private key along with an X509 v3 certificate that contains the public half of the key pair.

6. The Kerberos SSP on the client computer sends the user's public key certificate to the KDC as pre-authentication data in its initial authentication request.

7. The KDC validates the certificate, extracts the public key, and then uses the public key to encrypt a logon session key. It returns the encrypted logon session key and a TGT to the client.

8. If the client owns the private half of the key pair, it can use the private key to decrypt the logon session key. Both the client and the KDC then use this logon session key in all future communications with one another.

    - All cryptographic operations that use these keys take place on the smart card.

The rest of the authentication process is the same as for a standard logon session.

As the removable security device is inserted in response to the initial prompt for user logon, the system should recognize the “smart card” device being inserted, and then prompt for the user PIN.

Just as with the 802.1X authentication configuration and testing, vendors should ensure that Windows Logon using a smart card functions properly to supplement the user authentication to the domain. Procedures for setting up Windows Logon using a smart card can be found on the Microsoft web site.

3.1.1. Windows Logon Test Process

These tests will be done using a local intranet rather than a corporate network or internet type of connection. The server will be directly connected to the client.

1. Ensure that the user is configured for “Smart Card Logon” within Active Directory on the server.

2. Install the removable security device and software if you have not already done so.

3. Install the “Smart Card User” certificate from the enterprise smart card enrollment station onto the removable security device which can be used for 802.1X authentication. The user name provided will have been registered on the server. This same certificate can be used for 802.1X authentication.

4. Reboot the system. At this point, Windows should provide a different logon prompt which indicates to either “Insert card or press Ctrl-Alt-Delete to begin”.

5. Plug in the removable security device. You should be prompted to enter the access PIN. If Windows is able to successfully validate the credentials stored on the removable security device, you'll be logged in to Windows.

3.2. Simple 802.1X Authentication Configuration

Windows XP provides native support for IEEE 802.1X (Wired LAN or Wireless LAN) authentication using a smart card. This is a simple intrinsic capability of the operating system that should be tested. The vendor should enable 802.1X authentication to use smart card authentication credentials and ensure that it works properly on a notebook computer.

The following steps can be used to configure a Windows XP notebook for 802.1X authentication using a smart card. The test platform should be preconfigured with 802.11 wireless LAN support and must have a PCMCIA wireless LAN card. Furthermore, a 802.11 network access point (AP) must be available that supports the 802.1X protocol. This AP is connected to a RADIUS server that also includes 802.1X support.

1. Ensure that the system has been restored to its pristine condition. This particular test will only be done with Windows XP.

2. Install the removable security device and its software support

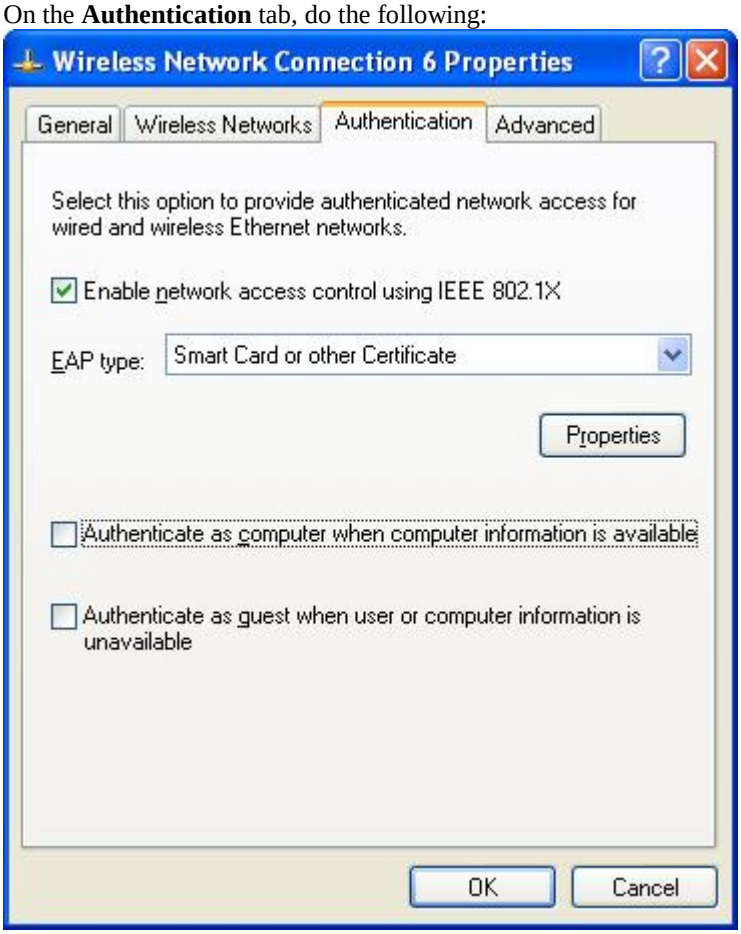
3. The following configuration options require system administrator privileges.

4. Open **Network Connections**

   a. To open Network Connections, click **Start**, click **Control Panel**, click **Network and Internet Connections**, and then click **Network Connections**.

5. Right-click the connection for to be enabled for IEEE 802.1x authentication, and then click **Properties**.

6. On the **Authentication** tab, do the following:



   a. To enable IEEE 802.1x authentication for this connection, select the **Network access control using IEEE 802.1X** check box. This check box is selected by default.

7. In **EAP type**, click the Extensible Authentication Protocol type to be used with this connection.

8. If **Smart Card or other Certificate** in **EAP type** is selected, additional properties can be configured by clicking on **Properties** and, in **Smart Card or other Certificate Properties**, do the following:

   a. To use the certificate that resides on the smart card for authentication, click **Use my smart card**.

   b. To verify that the server certificate presented to your computer is still valid, select the **Validate server certificate** check box, specify whether to connect only if the server resides within a particular domain, and then specify the trusted root certification authority.

   c. To use a different user name when the user name in the smart card or certificate is not the same as the user name in the domain to which the user is logging on, select the **Use a different user name for the connection** check box.

9. To specify whether the computer should attempt authentication to the network if a user is not logged on and/or if the computer or user information is not available, do the following:

a. To specify that the computer attempt authentication to the network if a user is not logged on, select the **Authenticate as computer when computer information is available** check box.

b. To specify that the computer attempt authentication to the network if user information or computer information is not available, select the **Authenticate as guest when user or computer information is unavailable** check box. This check box is selected by default.

- To configure settings on the **Authentication** tab, you must be a member of the local Administrators group.

- For wired and wireless network connections, the settings in the **Authentication** tab apply to the network to currently connected. If the current connection is to a wireless network, the name of the network can be verified by clicking the **Wireless Networks** tab. The name of the network will appear in **Visible networks** and **Preferred networks**, and it will be preceded by an icon with a circle around it.

Once configured for 802.1X authentication using a smart card, connection (either wired or wireless) automatically prompts the user to insert the smart card. Once the smart card insertion is detected, the user is prompted to enter the access PIN to allow the credentials to be retrieved and checked.

For additional information on 802.1X authentication using Windows XP, refer to Microsoft's web site and search for 802.1X.

3.2.1. 802.1X Authentication Test Process

802.1X authentication testing can be done on the same platform for the same removable security device with the same configuration that was used for Windows logon.

1. Plug in the wireless LAN card into the PCMCIA slot of the Banias reference system.

2. Windows will detect and configure the card.

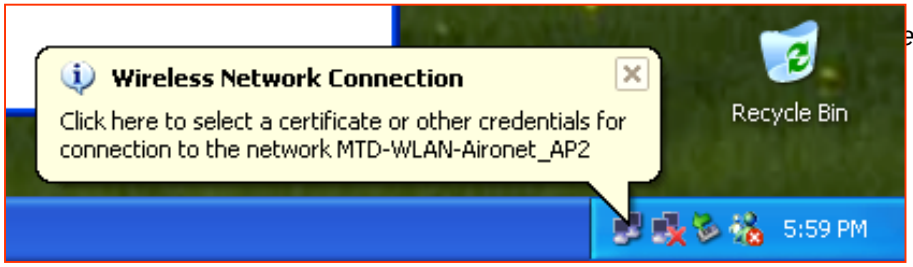
3. Configure the system for 802.1X as described above.

4. Install the removable security device and software if you have not already done so.

5. Install device software support onto the Windows .NET server to allow provisioning of the device with the user certificate.

6. Install the "Smart Card User" certificate from the Windows .NET enterprise smart card enrollment station onto the removable security device which can be used for 802.1X authentication. The user name provided will have been registered on the server. This same certificate can be used for Windows logon.

7. Windows will recognize that the connection requires 802.1X authentication using a certificate found on a "smart card". You will be prompted for the credentials to log onto the wireless LAN as shown in the picture below.



8. Click on that "Wireless Network Connection" balloon to be prompted to insert the security device containing the authentication certificate. You will be prompted to enter the PIN to access the device.

9. Open the Network Connections window (if it is not already open) to view connection status. When the device has been successfully authenticated using the certificate stored on the removable security device, the connection will be given a valid IP address.

3.3. Microsoft Outlook Email: Digitally Signed

NOTE:

- The process to acquire a new digital ID that each product test will require using Outlook Email is very time consuming and cannot be guaranteed to occur in a given time frame. VeriSign's web site says that it can take up to one hour to receive the response.

Preparation:

- Start by restoring the system using the fresh state. The system should include the OS, Microsoft Office, and the network and email configuration.
  - Set up email account to be used (such as Hotmail or your own corporate email account)
- Install Removable security device on the platform.

Get a digital ID for the account to be stored on the device:

1. With platform connected to the internet with valid Outlook XP email account.

2. Go to the VeriSign web site to download a free digital ID (http://digitalid.VeriSign.com/cgi-bin/haydn.exe?VHTML_FILE=client/outlook/OESEnrollFree.htm&name=&email=

   You can type that path directly, or go through this process:

   a. Run Microsoft Outlook

   b. Select from menu: Tools >> Options

   c. Open Security tab

d. Click on “Setup Secure Email” button. In Outlook XP, the button is labeled “Get a Digital ID”.

3. In the VeriSign web site, select the “60-day free trial”

4. The enrollment form includes 4 steps

    a. Enter Digital ID name. Each Digital ID Name has to be unique (you can’t use the same one everytime) so you may need to enter a number after each digital ID to make it unique.

    b. Enter Email address

    c. Enter Challenge Phrase such as “PortlandBlazers”

    d. Select the encryption device from the pull down list. You should see the CSP name for the encryption device being tested.

    e. Fill in the rest of the form.

    f. Remove the smart card device from the system (remove USB security token or pull the smart card from the reader). This is to make sure that when you finish the form, it actually tries to write to the security device rather than accidentally writing to one of the standard Microsoft software CSPs.

    g. Once you hit “Accept” at the bottom of the form, it’ll request a key to be generated from your crypto device.

    h. It should ask you to insert the device/card. Once connected, press Enter.

    i. It should then prompt you for the PIN to access the device.

    j. You then wait for the email response telling you to pick up your digital certificate from VeriSign.

5. The email response from VeriSign can come relatively quickly (within a few minutes for the experiments that I have done). It includes the path and password to pick up the digital ID. The VeriSign instructions do indicate that the email response can take up to an hour to receive. Suggestion: if the email response doesn’t come within 5 or 10 minutes, submit another request for a digital certificate with a new “digital ID name” but using the same email address.

6. Once you have the email from VeriSign, it gives directions on how to install the certificate. Follow those instructions to install the certificate to your security device. It should prompt you to enter your access PIN.

7. Add this digital signature to your email as follows:

    a. Within Outlook, go to Tools >> Options >> Security tab

    b. Check the box for “Add digital signature to outgoing messages”

    c. Click on the “Settings” button

d. "Choose" the certificate from the one you have downloaded to the security device. Do this for both "Signing Certificate" and "Encryption Certificate". You should see the digital certificate name that was used when you filled out the VeriSign request form.

e. Make sure the box is checked for "Send these certificates with signed messages"

f. Exit the selection box by pressing OK.

8. Generate an email (send it to your self). As you hit "send", the security device should prompt for an access PIN so that it can retrieve the digital ID.

9. You will see the email in your Inbox with a red "certificate" icon. Open the email, and open the certificate by clicking on the red icon. Verify the certificate is the one you expected.

3.4. Application Testing Checklist

Vendor Representative		
Product Name / Number		
Device Driver Name / Number		
Bus Interface (USB, PCMCIA)		

Table 2: Application Testing Checklist

	Guideline	Functions as Expected Windows 2000	Functions as Expected Windows XP
1.	Digitally signed email can be created using credentials stored and retrieved from the removable security device.	Yes No	Yes No
2.	Device can be easily installed and configured to provide 802.1X authentication credentials	NA	Yes No
3.	Device can be easily installed and configured to provide Windows logon credentials.	NA	Yes No

Comments:

4. Power Management, Power Consumption, and Performance

Power management and power consumption measurements of the device are to be measured on the Banias Customer Reference Board. Vendors may have other development environments that provide a mechanism to measure power consumption of the particular product under evaluation. The intent is to measure the power drawn by the device, not to include any of the PCMCIA or USB circuitry of the platform.

Software tools need to be installed on the system including:

- Both Windows 2000 and Windows XP.
- CSPTESTSUITE.EXE – Microsoft tool for testing CSP. Used to perform crypto activities on the device. Other vendor software tools may be used to exercise the crypto functions of the device. Using a simple "format smart card" utility provides adequate functions for these power measurements.

The removable security device and its support software should be installed before running these tests.

4.1. Power Management Testing

The major difference between a notebook computer and a desktop computer lie in the notebook's ability to move from place to place running off battery power. As such, it is important that removable security devices be tested with special emphasis done relative to power management aspects of the computer and interactive effects on the device.

4.1.1. ACPI D-state Testing

Removable security devices should support ACPI D-states as outlined in ACPI 2.0 specifications.

All USB devices must report support D-states D0 and D3 to be USB compliant. It is expected that all these devices not only report D-state support, but also manage transitions to lower power D-states when idle.

Using Windows Device Manager (with Windows XP only)

1. Click **Start**, click **Run**, type cmd.exe, and then press ENTER.
2. Type **set DEVMGR_SHOW_DETAILS=1** and then press ENTER.
3. Type **start devmgmt.msc** and then press ENTER. This will start up the device manager application.
4. In Device Manager, the properties for a device should now provide a **Details** tab that contains additional information about the device.
5. Select the device by name, usually under the "smart card reader" category.

6. Open the Details tab.

7. Within the pull down menu, select **Power Capabilities** Device capabilities should include multiple D-states.

8. Within the pull down menu, select **Power State Mappings** to show how these device power states are related to system power states.

4.1.2. ACPI S-state Testing

After installation and configuration of the removable security device, the system should be able to transition into Suspend and then resume without adversely impacting the functionality of the system and/or the device.

1. Ensure that the platform will enter and resume from Standby *before* configuring the removable security device.
   i. Start -> Shutdown -> Standby from the menu.
   ii. System should successfully go into sleep mode.
   iii. Repeat with system running with AC power *and* battery power if possible.
2. Install and configure removable security device
3. Ensure that it is functioning properly by using vendor tools to read configuration data from the device.
4. While the system is running with AC power plugged in, put the system into Suspend
   i. Start -> Shutdown -> Standby from the pulldown menu.
   ii. System should successfully go into sleep mode.
5. Use the system hardware buttons to resume from standby
6. Check functionality of the platform by launching an application from the system menu. This just ensures that an application can be started.
   i. Start -> Programs -> Accessories -> System Tools -> System Information
   ii. Exit the application.
7. Use vendor utility to access the removable security device.
8. Repeat this process at least 3 times, making sure the system is functional after each resume.
9. If system is capable of running using battery power, repeat steps 4-8 with the system using battery power.

4.1.3. ACPI C-state Testing

Basic functional testing of platform ACPI C-states and S-states with the removable security device in the notebook PC. This evaluation to be done only on system running Windows XP.

In order to determine C-state transitions, Windows XP provides a software tool called "PERFMON" which can be configured to display C-state changes.

1. Remove the removable security device from the USB or PCMCIA connection.

2. Click on the Windows "Start" button, and select "RUN"

3. Type PERFMON.EXE and press <Enter>.

4. After starting PERFMON, remove the default configured items.

5. Configure the display (right click on the graph) and add C3-state display to show. It may also be interesting to add C0 and Processor Idle.

6. Simply watch C-state transitions without the USB device plugged in, then plug in the USB device. After the USB device has determine that it is inactive, and has signaled the host that it can be powered down, you should be able to subsequently see the same type of C-state transitions as were seen before plugging the device into USB.

4.2. Power Consumption

Simple power measurements are to be recorded for the following scenarios listed below.

4.2.1. Smart card Readers: Power

Power measurements for smart card readers that do not provide cryptographic functions are to be done using a Gemsafe 8k smart card using the Gemsafe software utilities to format that smart card. Power measurement is to be taken for a period of 1 minute.

Power is to be measured at the connection point of the device (USB or PCMCIA). No other devices should be connected externally to those attach points.

- For PCMCIA power measurements, a modified PCMCIA adapter is available that should be connected to the NetDAQ or other similar power data capture device. Power consumption data should be gathered for the time duration specified in each step below, not just a single instantaneous reading.

- For USB devices, power data will be collected from the development board through the NetDAQ. Power measurements should be gathered for the time duration specified in each step below, not just a single instantaneous reading.

- Power data is recorded only for the device, and not for the platform as a whole.

Steps:

1. Restore the system to original state.

2. **Baseline power reading**: Power measurement should be taken *before connecting* the removable security device. Power should be measured at the attach point (USB or PCMCIA). It is assumed that no power draw should be found.

3. Install the device and it's software

4. Device power: The removable security device should be attached and software installed/configured. Once the system is stable (all drivers installed, no more system messages, …), and no applications are running, power should be measured at the attach point for the device (USB or PCMCIA).

5. Open a DOS box

6. Run the CSPTESTSUITE *N* times using the "-n" option. The Banias reference platform is connected to the NetDAQ data capture device such that power data can be recorded while CSPTESTSUITE is executing. Supply the parameter to allow this utility to run for ~5 minutes to get an average power consumption.

Power measurement with device active: power measurement is taken with device attached/plugged in (smart card inserted into the reader). Start data gathering when application CSPTESTSUITE starts and end power data gathering when test application stops. Alternate software tools can be used to exercise the crypto functions of the device. Minimum time for power data to be gathered should be no less than 1 minute.

4.2.2. USB Security Tokens and other Removable Security Devices: Power

USB security tokens and other removable security devices are to have power measured with the following guidelines. Power measurements are to be recording while the CSPTESTSUITE application is running to determine an average power during the run of the application. Five minutes running this application should be sufficient to get an average power consumption for the device. Supply the parameters to CSPTESTSUITE to allow it to run for ~5 minutes. Alternate software tools can be used to exercise the crypto functions of the device. Minimum time for power data to be gathered should be no less than 1 minute.

Power is to be measured at the connection point of the device (USB or PCMCIA). No other devices should be connected externally to those attach points.

- For PCMCIA power measurements, a modified PCMCIA adapter is available that should be connected to the NetDAQ. Power consumption data should be gathered for the time duration specified in each step below, not just a single instantaneous reading.

- For USB devices, power data will be collected from the development board through the NetDAQ. Power measurements should be gathered for the time duration specified in each step below, not just a single instantaneous reading.

- Power data is recorded only for the device, and not for the platform as a whole.

Steps:

1. Restore the system to original state.

2. **Baseline power reading**: Power measurement should be taken *before connecting* the removable security device. Power should be measured at the attach point (USB or PCMCIA). It is assumed that no power draw should be found.

3. Install the device and it's software

4. Device power: The removable security device should be attached and software installed/configured. Once the system is stable (all drivers installed, no more system messages, …), and no applications are running, power should be measured at the attach point for the device (USB or PCMCIA).

5. Open a DOS box

6. Run the CSPTESTSUITE *N* times using the "-n" option. The Banias reference platform is connected to the NetDAQ data capture device such that power data can be recorded while CSPTESTSUITE is executing. Supply the parameter to allow this utility to run for ~5 minutes to get an average power consumption.

7. Power measurement with device active: power measurement is taken with device attached/plugged in (smart card inserted into the reader). Start data gathering when application CSPTESTSUITE starts and end power data gathering when test application stops.

4.3. Performance

The purpose of gathering performance metrics is to provide comparison data when looking at multiple products in a similar category. Some products may have a wide range of performance characteristics when compared to other products.

Performance of crypto functions is to be measured by executing CSPTESTSUITE with the "-f" parameter. This provides data for how long it takes a device to perform the test. CSPTESTSUITE reports this data in its results log. This data is to be recorded in Table 4 below.

Performance measurements should be performed with smart card readers using a standard Gemplus, Schlumberger, or Infineon smart card.

Alternatively, performance to format a standard GemSafe 8k card can be done.

4.4. Power Management, Power Consumption, and Performance Checklists

Vendor Representative		
Product Name / Number		
Device Driver Name / Number		
Bus Interface (USB, PCMCIA)		

Table 3: ACPI State Verification

Guideline	Functions as Expected Windows 2000	Functions as Expected Windows XP
1. Device and supporting software provides ACPI D-state support to reduce power consumption.	NA	Yes No
2. Device and supporting software does not block ACPI S-state transitions	Yes No	Yes No
3. Device and supporting software allows host processor to enter C3 state as measured in Windows XP using "Perfmon" utility.	NA	Yes No

Comments:

Table 4: Power and Performance Measurements

Description	Windows 2000	Windows XP
1. **Baseline power reading**: Power measurement should be taken *before connecting* the removable security device. Power should be measured at the attach point (USB or PCMCIA). It is assumed that no power draw should be found.		
2. Device power: The removable security device should be attached and software installed/configured. Once the system is stable (all drivers installed, no more system messages, …), and no applications are running, power should be measured at the attach point for the device		

Description	Windows 2000	Windows XP
(USB or PCMCIA).		
3. Power measurement with device active: power measurement is taken with device attached/plugged in (smart card inserted into the reader). Start data gathering when application CSPTESTSUITE starts and end power data gathering when test application stops.		
a. Peak Power Reading		
b. Average Power Reading		
4. Performance measurement. Smart card used: _____ Process / application used: _____		

Comments:

5. Low Level Functional Testing

Low level functional testing is to be performed to validate functionality of the device in accordance to industry standard interfaces.

Smart card readers or other removable security devices that do not have intrinsic crypto functions are not expected to run these tests.

5.1. Microsoft CAPI Test Suite

Microsoft Cryptographic API is to be tested for conformance to Microsoft standards using Microsoft's CAPITESTSUITE software test tool.

Steps:

1. Restore system to virgin operating environment. Evaluation is to be run for both Windows 2000 and Windows XP.

2. Install hardware device and support software (device driver and CAPI CSP)

3. Ensure that the device has a CSP registered with the system by running CAPITESTSUITE within a DOS box without any command line parameters.

4. Use CAPITESTSUITE to test all CSP functions by running the tool

    a. Ensure that CSP is signed.

    b. Functional testing – test all CAPI interfaces using CAPITESTSUITE

5.2. Low Level Functional Test Checklist

Vendor Representative		
Product Name / Number		
Device Driver Name / Number		
Bus Interface (USB, PCMCIA)		

CAPI CSP DLL Name	

Version Information	

Table 5: Low Level CAPI Testing Results

Comments:

6. Source Code for PKCSVER.EXE

```
#include <windows.h>
#include <stdio.h>
#pragma pack(push, cryptoki, 1)
#include "pkcs11.h"
#pragma pack(pop, cryptoki)

int main(int argc, char *argv[])
{
    HINSTANCE hDll;
    CK_FUNCTION_LIST Functions;
    CK_FUNCTION_LIST_PTR pFunctions = NULL;
    CK_C_INITIALIZE_ARGS CkInitArgs;
    CK_INFO CkInfo;
    CK_BBOOL bInitialized = FALSE;
    CK_RV rv = CKR_FUNCTION_NOT_SUPPORTED;
    int verMajor, verMinor;

    // initialize
    CkInitArgs.CreateMutex = NULL;
    CkInitArgs.DestroyMutex = NULL;
    CkInitArgs.LockMutex = NULL;
    CkInitArgs.UnlockMutex = NULL;
    CkInitArgs.flags = CKF_OS_LOCKING_OK;
    CkInitArgs.pReserved = NULL;
    CkInfo.cryptokiVersion.major = 0;
    CkInfo.cryptokiVersion.minor = 0;

    // check dll name on cmdline
    if (argc < 2)
    {
       printf("\nERROR: Missing DLL name cmdline parameter\n");
       return 100;
    }

    // load the dll
    hDll = LoadLibrary(argv[1]);
    if (hDll == NULL)
    {
       printf("\nERROR: Cannot load DLL %s\n", argv[1]);
       return 104;
    }

    // try to get directly the functions
    Functions.C_Initialize = (CK_C_Initialize) GetProcAddress (hDll,
"C_Initialize");
    Functions.C_Finalize = (CK_C_Finalize) GetProcAddress (hDll,
"C_Finalize");
    Functions.C_GetInfo = (CK_C_GetInfo) GetProcAddress (hDll, "C_GetInfo");

    // if successful, call them
```

```
    if (Functions.C_GetInfo != NULL)
    {
       if (Functions.C_Initialize != NULL)
       {
          Functions.C_Initialize(&CkInitArgs);
          bInitialized = TRUE;
       }
       rv = Functions.C_GetInfo(&CkInfo);
       if (rv == CKR_OK && Functions.C_Finalize != NULL)
          Functions.C_Finalize(NULL);
    }

    if (rv != CKR_OK)                  // try the other way...
    {
       // need function GetFunctionList
       Functions.C_GetFunctionList = (CK_C_GetFunctionList) GetProcAddress
(hDll, "C_GetFunctionList");
       if (Functions.C_GetFunctionList == NULL)
       {
          FreeLibrary(hDll);
          printf("\nERROR: Missing function in DLL\n");
          return 108;
       }

       // get cryptoki function addresses
       rv = Functions.C_GetFunctionList(&pFunctions);
       if (rv != CKR_OK || pFunctions == NULL)
       {
          FreeLibrary(hDll);
          printf("\nERROR: No functions list\n");
          return 112;
       }
       if (pFunctions->C_Initialize == NULL ||
           pFunctions->C_GetInfo == NULL)
       {
          FreeLibrary(hDll);
          printf("\nERROR: Missing function in DLL\n");
          return 116;
       }

       // call functions
       if (!bInitialized)
          pFunctions->C_Initialize(&CkInitArgs);
       rv = pFunctions->C_GetInfo(&CkInfo);
       if (pFunctions->C_Finalize != NULL)
          pFunctions->C_Finalize(NULL);
       if (rv != CKR_OK)
       {
          FreeLibrary(hDll);
          printf("\nERROR: C_GetInfo RV=%x\n", rv);
          return 120;
       }
    }

    // report
    verMajor = (int) CkInfo.cryptokiVersion.major;
    verMajor &= 0xff;
    verMinor = (int) CkInfo.cryptokiVersion.minor;
    verMinor &= 0xff;
```

```
    if (verMinor == 1)
       printf("\nLibrary %s supports Cryptoki version %d.01\n",
              argv[1], verMajor);
    else
       printf("\nLibrary %s supports Cryptoki version %d.%d\n",
              argv[1], verMajor, verMinor);

    FreeLibrary(hDll);
    return 0;
}
=================== end source ==========================
```